CLAIMS:

1. A method for network protocol filtering of a packet, comprising:

determining packet type for the packet;

obtaining packet information for the packet;

determining whether the packet information is in a table;

responsive to the packet information being in the table, obtaining an index from the table; and

storing the index in a data structure in association with the packet.

2. The method, according to claim 1, further comprising:

determining whether the packet is for a new connection; and

responsive to the packet not being for the new connection, the determining whether the packet information is in the table.

3. The method, according to claim 2, wherein the packet type is a Transmission Control Protocol type.

4. The method, according to claim 1, wherein the packet type is a User Datagram Protocol type.

5. The method, according to claim 1, wherein the packet information is a five-tuple including source and destination addresses, source and destination ports, and a packet type identifier.

6. The method, according to claim 1, wherein the packet type is a Generic Routing Encapsulation type.

7. The method, according to claim 6, wherein the packet information is a five-tuple including source and destination addresses, an apportioned Generic Routing Encapsulation identifier, and a packet type identifier.

8. The method, according to claim 1, wherein the packet type is an Internet Protocol Security type.

9. The method, according to claim 8, wherein the packet information is a five-tuple including source and destination addresses, an apportioned security parameter string, and a packet type identifier.

10. The method, according to claim 1, wherein the table is a connection table and the index is to a network address translation table.

11. The method, according to claim 1, wherein the table is a network address translation table and the index is to a connection table.

12. The method, according to claim 1, further comprising:

using the index to identify another index; and

storing the other index in another data structure in association with the packet.

13. The method, according to claim 12, wherein the other index is to an address resolution table.

14. A method for inbound network address translation packet filtering, comprising:

obtaining a packet;

determining whether type of the packet is one of a Transmission Control Protocol, a User Datagram Protocol, a Generic Routing Encapsulation, an Internet Protocol Security and an Internet Control Message Protocol type;

if the type is the Transmission Control Protocol type, determining if the

packet is an initial packet for a connection;

if the type is the Transmission Control Protocol type and the packet is for an existing connection or if the type is one of the User Datagram Protocol type, the Generic Routing Encapsulation type and the Internet Protocol Security type,

obtaining packet information from the packet;

determining whether the packet information is in a first table;

responsive to the packet information being in the first table, obtaining a first index from the first table, the first index for a second table;

storing the first index in a data structure associated with the packet;

obtaining a second index from the second table responsive to the first index; and

storing the second index in the data structure;

if the type is the Internet Control Message Protocol type, determining whether the Internet Control Message Protocol type is on a list of Internet Control Message Protocol types;

obtaining a third index from one of the first table and the second table, the third index to a third table; and

storing the third index in the data structure.

15. The method, according to claim 14, wherein the data structure is for a plurality of canonical frame headers.

16. The method, according to claim 14, wherein the first table is a network address translation table.

17. The method, according to claim 16, wherein the second table is a connection table.

18. The method, according to claim 17, wherein the third table is an address resolution table.

19. The method, according to claim 18, further comprising:

checking validity of layers of the packet;

checking Internet Protocol options for the packet; and

determining whether the packet is a fragment.

20. The method, according to claim 19, further comprising determining whether the network address translation is supported by a network processing unit.

21. A method for inbound network address translation packet filtering, comprising:

obtaining a packet;

determining whether type of the packet is one of a Transmission Control Protocol, a User Datagram Protocol, a Generic Routing Encapsulation, an Internet Protocol Security type;

if the type is the Transmission Control Protocol type, determining if the packet is an initial packet for a connection;

if the type is the Transmission Control Protocol type and the packet is for an existing connection or if the type is one of the User Datagram Protocol type, the Generic Routing Encapsulation type and the Internet Protocol Security type;

obtaining packet information from the packet;

determining whether the packet information is in a first table;

responsive to the packet information being in the first table, obtaining a first index from the first table, the first index for a second table;

storing the first index in a data structure associated with the packet;

obtaining a second index from the second table responsive to the first index;

storing the second index in the data structure;

obtaining a third index from one of the first table and the second table, the third index to a third table; and

storing the third index in the data structure.

22. The method, according to claim 21, wherein the data structure is for a plurality of canonical frame headers.

23. The method, according to claim 21, wherein the first table is a network address translation table.

24. The method, according to claim 23, wherein the second table is a connection table.

25. The method, according to claim 24, wherein the third table is an address resolution table.

26. A method for inbound network address translation packet filtering, comprising:

obtaining a packet;

determining whether type of the packet is one of a Transmission Control Protocol and a User Datagram Protocol;

if the type is the Transmission Control Protocol type, determining if the packet is an initial packet for a connection;

if the type is the Transmission Control Protocol type and the packet is for an existing connection or if the type is the User Datagram Protocol type,

obtaining packet information from the packet;

determining whether the packet information is in a first table;

responsive to the packet information being in the first table, obtaining a first index from the first table, the first index for a second table;

storing the first index in a data structure associated with the packet;

obtaining a second index from the second table responsive to the first index;

storing the second index in the data structure;

obtaining a third index from one of the first table and the second

table, the third index to a third table; and

storing the third index in the data structure.

27. The method, according to claim 26, wherein the data structure is for a plurality of canonical frame headers.

28. The method, according to claim 26, wherein the first table is a network address translation table.

29. The method, according to claim 28, wherein the second table is a connection table.

30. The method, according to claim 29, wherein the third table is an address resolution table.

31. A method for outbound packet filtering, comprising:

obtaining a packet;

determining whether an incoming interface for the packet is running network address translation;

if the incoming interface is running the network address translation,

obtaining a first index from a data structure associated with the packet; and

obtaining packet information in a first table using the first index;

determining whether type of the packet is one of a Transmission Control Protocol, a User Datagram Protocol, a Generic Routing Encapsulation, an Internet Protocol Security and an Internet Control Message Protocol type;

if the type is the Transmission Control Protocol type, determining if the packet is an initial packet for a connection;

if the type is the Transmission Control Protocol type and the packet is for an existing connection or if the type is the User Datagram Protocol type,

obtaining the packet information from the packet;

determining whether the packet information is in a second table;

responsive to the packet information being in the second table, obtaining a second index from the second table;

storing the second index in the data structure;

checking whether the packet is the Transmission Control Protocol type; and

responsive to the packet being the Transmission Control Protocol type, checking for a Transmission Control Protocol state error of the packet;

if the type is the Generic Routing Encapsulation type or the Internet Protocol Security type,

obtaining packet information from the packet;

determining whether the packet information is in the second table;

responsive to the packet information being in the second table, obtaining the second index from the second table; and

storing the second index in the data structure;

if the type is the Internet Control Message Protocol type, determining whether the Internet Control Message Protocol type is on a list of Internet Control Message Protocol types;

if the type is not the Internet Control Message Protocol type,

determining if the outgoing interface is running the network address translation;

responsive to the outgoing interface running the network address translation,

obtaining the second index from the data structure; and

obtaining the packet information from the first table using the second index.

32. The method, according to claim 31, wherein the data structure is for a plurality of canonical frame headers.

33. The method, according to claim 31, wherein the first table is a network address translation table.

34. The method, according to claim 33, wherein the second table is a connection table.

35. The method, according to claim 34, wherein the packet information is a five-tuple of packet information.

36. The method, according to claim 31, wherein the packet information is a five-tuple including source and destination addresses, source and destination ports, and a packet type identifier.

37. The method, according to claim 31, wherein the packet type is a Generic Routing Encapsulation type.

38. The method, according to claim 37, wherein the packet information is a five-tuple including source and destination addresses, an apportioned Generic Routing Encapsulation identifier, and a packet type identifier.

39. The method, according to claim 31, wherein the packet type is an Internet Protocol Security type.

40. The method, according to claim 39, wherein the packet information is a five-tuple including source and destination addresses, an apportioned security parameter string, and a packet type identifier.

41. The method, according to claim 31, further comprising:

checking validity of layers of the packet;

checking Internet Protocol options for the packet; and

determining whether the packet is a fragment.

42. The method, according to claim 31, further comprising determining whether the network address translation is supported by a network processing unit.

43. A method for outbound packet filtering, comprising:

obtaining a packet;

determining whether an incoming interface for the packet is running network address translation;

if the incoming interface is running the network address translation,

obtaining a first index from a data structure associated with the packet; and

obtaining packet information in a first table using the first index;

if the incoming interface is not running the network address translation, determining whether type of the packet is one of a Transmission Control Protocol, a User Datagram Protocol, a Generic Routing Encapsulation, and an Internet Protocol Security type;

if the type is the Transmission Control Protocol type, determining if the packet is an initial packet for a connection;

if the type is the Transmission Control Protocol type and the packet is for an existing connection or if the type is the User Datagram Protocol type,

obtaining the packet information from the packet;

determining whether the packet information is in a second table;

responsive to the packet information being in the second table, obtaining a second index if present from the second table;

storing the second index if present in the data structure;

checking whether the packet is the Transmission Control Protocol type; and

responsive to the packet being the Transmission Control Protocol type, checking for a Transmission Control Protocol state error of the packet;

if the type is the Generic Routing Encapsulation type or the Internet

Protocol Security type,

obtaining packet information from the packet;

determining whether the packet information is in the second table;

responsive to the packet information being in the second table, obtaining the second index if present from the second table; and

storing the second index if present in the data structure;

determining if the outgoing interface is running the network address translation;

responsive to the outgoing interface running the network address translation,

obtaining the second index from the data structure; and

obtaining the packet information from the first table using the second index.

44. The method, according to claim 43, wherein the data structure is for a plurality of canonical frame headers.

45. The method, according to claim 43, wherein the first table is a network address translation table.

46. The method, according to claim 45, wherein the second table is a connection table.

47. A method for outbound packet filtering, comprising:

obtaining a packet;

determining whether an incoming interface for the packet is running network address translation;

if the incoming interface is running the network address translation,

obtaining a first index from a data structure; and

obtaining packet information in a first table using the first index;

if the incoming interface is not running the network address translation, determining whether type of the packet is one of a Transmission Control Protocol and a User Datagram Protocol type;

if the type is the Transmission Control Protocol type, determining if the packet is an initial packet for a connection;

if the type is the Transmission Control Protocol type and the packet is for an existing connection or if the type is the User Datagram Protocol type,

obtaining the packet information from the packet;

determining whether the packet information is in a second table;

responsive to the packet information being in the second table, obtaining a second index if present from the second table;

storing the second index if present in the data structure;

checking whether the packet is the Transmission Control Protocol type; and

responsive to the packet being the Transmission Control Protocol type, checking for a Transmission Control Protocol state error of the packet;

determining if the outgoing interface is running the network address translation;

responsive to the outgoing interface running the network address translation,

obtaining the second index from the data structure; and

obtaining the packet information from the first table using the second index.

48. The method, according to claim 47, wherein the data structure is for a plurality of canonical frame headers.

49. The method, according to claim 47, wherein the first table is a network address translation table.

50. The method, according to claim 49, wherein the second table is a connection table.

51. A method for outbound packet filtering, comprising:

obtaining a packet;

determining whether an incoming interface for the packet is running network address translation;

if the incoming interface is running the network address translation,

obtaining a first index from a data structure associated with the packet;

obtaining packet information in a table using the first index;

checking whether the packet is the Transmission Control Protocol type; and

responsive to the packet being the Transmission Control Protocol type, checking for a Transmission Control Protocol state error of the packet;

determining if the outgoing interface is running the network address translation; and

responsive to the outgoing interface running the network address translation,

obtaining a second index from the data structure; and

obtaining the packet information from the table using the second index.

52. The method, according to claim 51, wherein the data structure is for a plurality of canonical frame headers.

53. The method, according to claim 51, wherein the table is a network address translation table.

54. The method, according to claim 53, wherein the first index is to a connection table, and wherein the second index is to the network address translation table.

55. An apparatus for network protocol filtering of a packet, comprising:

means for determining packet type for the packet;

means for obtaining packet information for the packet;

means for determining whether the packet information is in a table;

means for obtaining an index from the table responsive to the packet information being in the table; and

means for storing the index in a header of the packet.

56. A signal-bearing medium containing a program, which when executed by a processor, causes execution of a network protocol packet filtering method comprising:

determining packet type for the packet;

obtaining packet information for the packet;

determining whether the packet information is in a table;

obtaining an index from the table responsive to the packet information being in the table; and

storing the index in a header of the packet.

57. A method for network address translating, comprising:

obtaining a packet for network address translation, the packet having a media access control header;

determining if a network processing unit is in a pass-through mode responsive for the packet; and

responsive to the network processing unit not being in the pass-through mode,

obtaining a media access control source address from the media

access control header is stored in an address resolution table;

determining whether an incoming interface is running network address translation; and

network address translation filtering the packet responsive to the incoming interface running the network address translation, the network address translation filtering including,

obtaining an address resolution table index from the packet.

58. The method, according to claim 57, wherein the pass-through mode is a firewall only mode.

59. The method, according to claim 57, further comprising:

determining whether the packet is for a multicast or broadcast frame;

determining whether the incoming interface equals an outgoing interface; and

reading control bits for the packet responsive to the media access control source address obtained.

60. The method, according to claim 59, further comprising:

determining protocol type of the packet; and

determining whether the protocol type is supported on the outgoing interface.

61. The method, according to claim 60, further comprising determining whether broadcasting or multicasting is invoked for the outgoing interface.

62. An apparatus for network address translating, comprising:

means for obtaining a packet for network address translation, the packet having a media access control header;

means for determining if a network processing unit is not in a pass-through mode responsive for the packet;

means for obtaining a media access control source address from the media access control header is stored in an address resolution table;

means for reading control bits for the packet responsive to the media access control source address obtained;

means for determining whether an incoming interface is running network address translation; and

means for network address translation filtering the packet responsive to the incoming interface running the network address translation, the means for network address translation filtering including means for obtaining an address resolution table index from the packet.

63. A signal-bearing medium containing a program which, when executed by a processor, causes execution of a method for network address translating comprising:

obtaining a packet for network address translation, the packet having a media access control header;

determining if a network processing unit is not in a pass-through mode responsive for the packet;

obtaining a media access control source address from the media access control header is stored in an address resolution table;

determining whether an incoming interface is running network address translation; and

network address translation filtering the packet responsive to the incoming interface running the network address translation, the network address translation filtering including obtaining an address resolution table index from the packet.

64. A method for outbound packet filtering, comprising:

determining if network address translation is running on an inbound interface;

responsive to network address translation running on the inbound interface,

obtaining a connection table index and a network address table index; and

translating local address packet information to public address packet information for a packet.

65. The method, according to claim 64, wherein the translating comprises obtaining the local address packet information and the public address packet information from a connection table and a network address translation table, respectively, respectively responsive to the connection table index and the network address table index.

66. The method, according to claim 65, further comprising:

determining if the packet is a Transmission Control Protocol ("TCP") packet;

responsive to the packet being a TCP packet,

checking validity of Internet Protocol options; and

checking TCP state for an error.